5

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/601,066	06/20/2003	Pekka Lehtinen	915-005.064	8887

4955 7590 05/04/2007

WARE FRESSOLA VAN DER SLUYS &
ADOLPHSON, LLP
BRADFORD GREEN, BUILDING 5
755 MAIN STREET, P O BOX 224
MONROE, CT 06468

EXAMINER	
MARCYES, JAY R	
ART UNIT	PAPER NUMBER
2109	

MAIL DATE	DELIVERY MODE
05/04/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/601,066	LEHTINEN, PEKKA	
	Examiner	Art Unit	
	Jay R. Marcyes	2109	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *20 June 2003*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-33* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-33* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☒ All b)☐ Some * c)☐ None of:

1.☒ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date *08/11/2005*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

Remarks

Examiner notes that applicant has invoked 35 U.S.C 112 6th paragraph by using "means-for" language in claims **1-4, 7, 22, 31-32**. This has been taken into account for prior art considerations.

Claim Rejections - 35 USC § 101

35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

Claims **33** are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter. The specification does not specifically define a computer program as being limited to a tangible computer readable medium such as a disk or EPROM. A reasonable interpretation could include an electromagnetic carrier wave (i.e., a signal) or the like. Such a signal is not a process, machine, manufacture, or composition of matter. Thus, the claims are directed to non-statutory subject matter.

The specification must be amended to include a specific and concise definition of article of manufacture and program usable medium. No new matter may be added.

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

Claims **1-4, 6-7, 10-13, 16-19, 21-22, 25-28, 31-33** are rejected under 35 U.S.C. 102(b) as being anticipated by Nishihara et al. (US Patent 6,026,427).

With respect to claim **1**, Nishihara et al. teaches a system comprising means for executing application sessions in an electronic device with one or more processors (*Figure 1*), and means for scheduling Resource Reservation Instances as well as the execution of substantially simultaneous application sessions (*Figure 4*), wherein the application session to be executed comprises one or more Activity Blocks in one or more Activity Block Containers (*column 3 lines 53-54; Thread is analogous to Activity Blocks and Process is analogous to Activity Block Containers*), and an execution order is specified for said Activity Blocks (*Column 3 lines 56-57*); the system further comprising resource type specific Resource Handlers for reserving resources for the application session (*Figure 3, Condition Variable Module 45*), Resource Allocation Manager for analyzing and saving the resource allocation situation (*Figure 3, Condition Variable Module 45*), Application Session Management and Scheduling means for selecting at least the next application session and Activity Block to be executed on the basis of said resource allocation situation (*Figure 3, Condition Variable Module 45; Figure 4, Semaphore 50*), executing means for executing the next Activity Block in the course of the selected application session (*Column 3, lines 15-16; Figure 1, Processor 12*), and the system is provided with a protocol connecting the Resource Handlers, Resource Allocation Manager, Application Session Management and Scheduling means and executing means, to control the execution order and to implement the transfer of

information between said Resource Handlers, Resource Allocation Manager, Application Session Management and Scheduling means, and executing means (*Figure 1, Interconnection System 16*).

With respect to claim **2**, Nishihara et al. further teaches comprising means for bookkeeping of the resource allocation situation, means for transmitting a first control message to an Activity Block to provide control information on the resource allocation to the Activity Block at the time of the initiation of the Activity Block, and means for transmitting a second control message at the time of the completion of the execution of the Activity Block to provide information about the resources reserved or released by the Activity Block to update the bookkeeping of the resource allocation situation after the completion of each Activity Block (*Figure 4, Semaphore 50*).

With respect to claim **3**, Nishihara et al. further teaches comprising means for an application session to reserve the resources needed by each Activity Block (*Figure 4, Waited Counter 52*), as well as to release them (*Figure 4, Signal Counter 51*), either directly from the resource type specific Resource Handlers or from the Resource Allocation Manager that enable the queuing of reservation request messages, on the basis of control parameters received in a first control message received from Application Session Management and Scheduling means (*Figure 4*).

With respect to claim **4**, Nishihara et al. further teaches comprising means for making the Resource Reservation Instances created on a request from the application session, via the use of second Control messages, dynamically available to different

Activity Block Containers involved in the execution of the session, as needed (*Figure 1, Interconnection System 16*).

With respect to claim **6**, Nishihara et al. further teaches comprising a Resource Instance Table per each Resource Handler to provide the resource allocation situation to said resource management and allocation means, and the synchronization of the Resource Allocation Manager with respect to the Resource Handlers is arranged to be determined so that substantially immediately after each execution turn of the Resource Handlers it is the turn of the Resource Allocation Manager, wherein the resource allocation situation is unambiguously known in the Resource Instance Tables regarding the latest changes occurred (*Figure 3, Condition Variable Module 45*).

With respect to claim **7**, Nishihara et al. further teaches wherein the synchronization of the Resource Allocation Manager with respect to the Application Session Management and Scheduling means is determined so that substantially immediately after each execution turn of the Resource Allocation Manager it is the turn of the Application Session Management and Scheduling means, wherein the resource allocation situation is unambiguously known regarding the latest changes occurred, and values can be determined by the Application Session Management and Scheduling means for the parameters of the control messages generated by it for the synchronization of the use of various types of Resource Reservation Instances (*Figure 3, Condition Variable Module 45*).

With respect to claim **10**, Nishihara et al. further teaches wherein the Activity Blocks of the application session are placed in one or more Activity Block Containers,

that Activity Blocks in any one of these Activity Block Containers are arranged to be executed temporally at different times, and in the presence of Activity Blocks that are intended to be executed substantially at the same time in the course of the session, they are placed in different Activity Block Containers (*column 3, lines 15-19*).

With respect to claim **11**, Nishihara et al. further teaches wherein for designing applications that are to be executed in the system, each Activity Block Container is furnished with an interface module at those points where the execution of an Activity Block or the Activity Block Container can be interrupted and it may be the turn of another OS task to be executed, thus enabling the sending and reception of Session Control Protocol messages to take place via this interface of the Activity Block Container without a need to deal with these messages of the protocol as part of the application design work (*Figure 3, Condition Variable Module 45; Figure 1, Interconnection System 16*).

With respect to claim **12**, Nishihara et al. further teaches wherein the Resource Handlers are equipped with an interface for transmitting information between each Resource Handler and the system, this interface being substantially independent of the application session and the resource type (*Figure 1, Interconnection System 16*).

With respect to claim **13**, Nishihara et al. further teaches comprising a dedicated Resource Instance Table in the use of each Resource Handler (*Figure 3, Condition Variable Module 45*), and the Resource Handlers are designed to be without intermediate delayed states, wherein the changes of the status data of individual

Resource Reservation Instances are stored in the Resource Instance Table of each Resource Handler (*Column 3, lines 40-60*).

With respect to claim **16**, Nishihara et al. teaches a method for executing application sessions in an electronic device with one or more processors for synchronizing Resource Reservation Instances as well as the execution of substantially simultaneous application sessions (*Column 1, lines 10-12*), wherein the application session to be executed comprises one or more Activity Blocks in one or more Activity Block Containers, and an execution order is determined for, said Activity Blocks (*Column 3, lines 13-15*), the method comprising at least the following steps:

- a resource management and allocation step for requesting and reserving resources for the application session (*Figure 5, T1 and T2*),

- a bookkeeping and analysis step for saving and analyzing the resource reservation situation (*Figure 5, T1 and T2*),

- a scheduling and selection step for selecting the next application session and Activity Block to be executed at least on the basis of said resource reservation situation (*Figure 5, T3*),

- an execution step for executing the next Activity Block in the course of the selected application session (*Figure 5, T4*),

wherein in the method, a communication protocol connecting said resource management and allocation step, bookkeeping and analysis step, scheduling and selection step, and the execution step are used to control the execution order and, if

necessary, to transfer information between said resource management and allocation step, bookkeeping and analysis step, scheduling and selection step, and execution step (*Figure 1, Interconnection System 16*).

With respect to claim **17**, Nishihara et al. further teaches wherein a bookkeeping of the resource allocation is maintained, and an first control message is transmitted to an Activity Block to provide control information on the resource allocation at the time of the initiation of the Activity Block, and a second Control message is returned by the Activity Block to provide information about the resources reserved or released by the Activity Block to update the bookkeeping of the resource allocation situation after the completion of each Activity Block (*Figure 4, Semaphore 50*).

With respect to claim **18**, Nishihara et al. further teaches wherein the resources needed by each Activity Block are reserved (*Figure 4, Waited Counter 52*) and released (*Figure 4, Signal Counter 51*) by the application session, either directly from resource type specific Resource Handlers or from the Resource Allocation Manager that enable the queuing of Reservation Request messages, on the basis of control parameters received in a first control message received from Application Session Management and Scheduling means (*Figure 4*).

With respect to claim **19**, Nishihara et al. further teaches wherein second control messages are used by the application session to dynamically assign Resource Reservation Instances to the use of different Activity Block Containers involved in the execution of the session, as needed (*Figure 4, Semaphore 50*).

With respect to claim **21**, Nishihara et al. further teaches wherein in the method, a Resource Instance Table is used per each Resource Handler to provide the resource allocation situation to said Resource Allocation Manager, and the synchronization of the bookkeeping and analysis step with respect to the resource management and allocation step of the Resource Handlers is determined so that substantially immediately after each execution turn of the Resource Handlers, it is the turn of the bookkeeping and analysis step, wherein the resource allocation situation is unambiguously known in the Resource Instance Tables regarding the changes occurred (*Figure 3, Condition Variable Module 45*).

With respect to claim **22**, Nishihara et al. further teaches wherein the scheduling of the scheduling and selection step with respect to the bookkeeping and analysis step is determined so that the scheduling and selection step is in turn substantially immediately after the execution of the bookkeeping and analysis step, wherein the resource allocation situation is unambiguously known in the Resource Allocation Table regarding the latest changes occurred, and values can be determined by the Application Session Management and Scheduling means for the parameters of the control messages generated by it for the synchronization of the use of various types of Resource Reservation Instances (*Figure 5; Figure 3, Condition Variable Module 45*).

With respect to claim **25**, Nishihara et al. further teaches wherein the Activity Blocks of the application session are placed in one or more Activity Block Containers, Activity Blocks in any one of these Activity Block Containers are executed temporally at different times, and in the presence of Activity Blocks that are intended to be executed

substantially at the same time in the course of the session, they are placed in different Activity Block Containers (*column 3, lines 15-19*).

With respect to claim **26**, Nishihara et al. further teaches wherein for designing applications that are to be executed in the system, each Activity Block Container is furnished with an interface module at those points where the execution of an Activity Block or Activity Block Container can be interrupted and it may be the turn of another Operating System task to be executed, thus enabling the sending and reception of Session Control Protocol messages to take place via this interface of the Activity Block Container without a need to deal with these messages of the Session Control Protocol as part of the application design work (*Figure 3, Condition Variable Module 45; Figure 1, Interconnection System 16*).

With respect to claim **27**, Nishihara et al. further teaches wherein the Resource Handlers are equipped with an interface for transmitting information between each Resource Handler of the system, this interface being substantially independent of the application session and the resource type (*Figure 1, Interconnection System 16*).

With respect to claim **28**, Nishihara et al. further teaches wherein a dedicated Resource Instance Table is in the use of each Resource Handler (*Figure 3, Condition Variable Module 45*), and the Resource Handlers are designed to be without intermediate delayed states, wherein the changes of the status data of individual Resource Reservation Instances are stored in the Resource Instance Table of each Resource Handler (*Column 3, lines 40-60*).

With respect to claim **31**, Nishihara et al. teaches an electronic device comprising means for executing application sessions, one or more processors, and means for scheduling Resource Reservation Instances as well as the execution of substantially simultaneous application sessions (*Column 1, lines 14-16*), wherein the application session to be executed comprises one or more Activity Blocks in one or more Activity Block Containers (*Column 3, lines 13-15*), and an execution order is determined for said Activity Blocks (*Column 3, line 57*); the electronic device further comprising resource type specific Resource Handlers for reserving resources for the application session (*Figure 4, Semaphore 50*), Resource Allocation Manager for analyzing and saving a resource allocation situation, Application Session Management and Scheduling means for selecting at least the next application session and Activity Block to be executed on the basis of said resource allocation situation (*Figure 3, Condition Variable Module 45*), executing means for executing the next Activity Block in the course of the selected application session (*Column 3, line 15*); and the electronic device is provided with a protocol connecting the Resource Handlers, Resource Allocation Manager, Application Session Management and Scheduling means and executing means, to control the execution order and to implement the transfer of information between said Resource Handlers, Resource Allocation Manager, Application Session Management and Scheduling means, and executing means (*Figure 1, Interconnection System 16*).

With respect to claim **32**, Nishihara et al. teaches an wireless communication device comprising means for executing application sessions, one or more processors,

and means for scheduling Resource Reservation Instances as well as the execution of substantially simultaneous application sessions (*Column 1, lines 14-16*), wherein the application session to be executed comprises one or more Activity Blocks in one or more Activity Block Containers (*Column 3, lines 13-15*), and an execution order is determined for said Activity Blocks (*Column 3, line 57*); the wireless communication device further comprising resource type specific Resource Handlers for reserving resources for the application session (*Figure 4, Semaphore 50*), Resource Allocation Manager for analyzing and saving a resource allocation situation, Application Session Management and Scheduling means for selecting at least the next application session and Activity Block to be executed on the basis of said resource allocation situation (*Figure 3, Condition Variable Module 45*), executing means for executing the next Activity Block in the course of the selected application session (*Column 3, line 15*); and the electronic device is provided with a protocol connecting the Resource Handlers, Resource Allocation Manager, Application Session Management and Scheduling means and executing means, to control the execution order and to implement the transfer of information between said Resource Handlers, Resource Allocation Manager, Application Session Management and Scheduling means, and executing means (*Figure 1, Interconnection System 16*).

With respect to claim **33**, Nishihara et al. teaches a software program comprising machine executable steps for executing application sessions in an electronic device with one or more processors for synchronizing Resource Reservation Instances as well

as the execution of substantially simultaneous application sessions (*Column 1, lines 10-12*), wherein the application session to be executed comprises one or more Activity Blocks in one or more Activity Block Containers, and an execution order is determined for said Activity Blocks (*Column 3, lines 13-15*), the software program further comprising machine performing at least the following steps: executable steps for

- a resource management and allocation step for requesting and reserving resources for the application session (*Figure 5, T1 and T2*),

- a bookkeeping and analysis step for saving and analyzing the resource reservation situation (*Figure 5, T1 and T2*),

- a scheduling and selection step for selecting the next application session and Activity Block to be executed at least on the basis of said resource reservation situation (*Figure 5, T3*),

- an execution step for executing the next Activity Block in the course of the selected application session (*Figure 5, T4*),

wherein the software program also comprises machine executable steps for using a communication protocol connecting said resource management and allocation step, bookkeeping and analysis step, scheduling and selection step, and the execution step to control the execution order and, if necessary, to transfer information between said resource management and allocation step, bookkeeping and analysis step, scheduling and selection step, and execution step (*Figure 1, Interconnection System 16*).

Claims **1-8, 10-14, 16-23, 25-29, 31-33** are rejected under 35 U.S.C. 102(b) as being anticipated by Alford et al. (US Patent Publication 2002/0065953 A1).

With respect to claim **1**, Alford et al. teaches a system comprising means for executing application sessions in an electronic device with one or more processors (*Figure 3, Processor 350*), and means for scheduling Resource Reservation Instances as well as the execution of substantially simultaneous application sessions (*Figure 10, Scheduling Function 1000*), wherein the application session to be executed comprises one or more Activity Blocks in one or more Activity Block Containers (*Figure 6, Application 611*), and an execution order is specified for said Activity Blocks (*Figure 2*); the system further comprising resource type specific Resource Handlers for reserving resources for the application session (*Figure 2, Scheduler 220*), Resource Allocation Manager for analyzing and saving the resource allocation situation (*Figure 2, Scheduler 220*), Application Session Management and Scheduling means for selecting at least the next application session and Activity Block to be executed on the basis of said resource allocation situation (*Figure 2, Scheduling Queue 215*), executing means for executing the next Activity Block in the course of the selected application session (*Figure 3, Processor 350*), and the system is provided with a protocol connecting the Resource Handlers, Resource Allocation Manager, Application Session Management and Scheduling means and executing means, to control the execution order and to implement the transfer of information between said Resource Handlers, Resource Allocation Manager, Application Session Management and Scheduling means, and executing means (*Figure 2, Internal Msg Source 206 and External Msg Source 205*).

With respect to claim **2**, Alford et al. further teaches comprising means for bookkeeping of the resource allocation situation, means for transmitting a first control message to an Activity Block to provide control information on the resource allocation to the Activity Block at the time of the initiation of the Activity Block, and means for transmitting a second control message at the time of the completion of the execution of the Activity Block to provide information about the resources reserved or released by the Activity Block to update the bookkeeping of the resource allocation situation after the completion of each Activity Block (*Figure 3, PTE Application Interface 330*).

With respect to claim **3**, Alford et al. further teaches comprising means for an application session to reserve the resources needed by each Activity Block (*Figure 2, Internal Msg Source 206 and External Msg Source 205*), as well as to release them (*Figure 2, Internal Msg Source 206 and External Msg Source 205*), either directly from the resource type specific Resource Handlers or from the Resource Allocation Manager that enable the queuing of reservation request messages, on the basis of control parameters received in a first control message received from Application Session Management and Scheduling means (*Figure 11*).

With respect to claim **4**, Alford et al. further teaches comprising means for making the Resource Reservation Instances created on a request from the application session, via the use of second Control messages, dynamically available to different Activity Block Containers involved in the execution of the session, as needed (*Figure 7*).

With respect to claim **5**, Alford et al. further teaches comprising an Operating System with scheduling functions (*Figure 3*), and for synchronizing the reservation,

release and other resource-related control from the Application Session Management and Scheduling means, Activity Block Containers, Resource Allocation Manager, and Resource Handlers (*Figure 3, PTE 330*), there is a Session Control Protocol composed of application-independent control messages and rules on their use, which is arranged during its operation to implement the synchronization and scheduling control of the execution of the Application Session Management and Scheduling means, the Activity Block Containers, the Resource Allocation Manager, as well as the Resource Handlers, on the basis of the task switching functions of the Operating System as well as the OS task priorities defined for the Application Session Management and Scheduling means, the Activity Block Containers, the Resource Allocation Manager, and the Resource Handlers (*Figure 5, API 540*).

With respect to claim **6**, Alford et al. further teaches comprising a Resource Instance Table per each Resource Handler to provide the resource allocation situation to said resource management and allocation means, and the synchronization of the Resource Allocation Manager with respect to the Resource Handlers is arranged to be determined so that substantially immediately after each execution turn of the Resource Handlers it is the turn of the Resource Allocation Manager, wherein the resource allocation situation is unambiguously known in the Resource Instance Tables regarding the latest changes occurred (*Figures 13a-b*).

With respect to claim **7**, Alford et al. further teaches wherein the synchronization of the Resource Allocation Manager with respect to the Application Session Management and Scheduling means is determined so that substantially immediately

after each execution turn of the Resource Allocation Manager it is the turn of the Application Session Management and Scheduling means, wherein the resource allocation situation is unambiguously known regarding the latest changes occurred, and values can be determined by the Application Session Management and Scheduling means for the parameters of the control messages generated by it for the synchronization of the use of various types of Resource Reservation Instances (*Figure 12*).

With respect to claim **8**, Alford et al. further teaches wherein an End State Module is placed at the end of each Activity Block to complete the execution of the block (*Figure 12, End 1270*), and a Waiting State Module is placed in the Activity Block Container holding the Activity Block (*Figure 9, Blocked 930*), and that the execution control of the Activity Block Container holding the Activity Block is arranged to generate a second Control message in the End State Module and to pause the execution in the Waiting State Module in order to wait for a first control message from the Application Session Management and Scheduling means, wherein the execution of the application session is temporarily interrupted regarding the current Activity Block Container (*Figure 11*).

With respect to claim **10**, Alford et al. further teaches wherein the Activity Blocks of the application session are placed in one or more Activity Block Containers, that Activity Blocks in any one of these Activity Block Containers are arranged to be executed temporally at different times, and in the presence of Activity Blocks that are

intended to be executed substantially at the same time in the course of the session, they are placed in different Activity Block Containers (*Figure 6*).

With respect to claim **11**, Alford et al. further teaches wherein for designing applications that are to be executed in the system, each Activity Block Container is furnished with an interface module at those points where the execution of an Activity Block or the Activity Block Container can be interrupted and it may be the turn of another OS task to be executed, thus enabling the sending and reception of Session Control Protocol messages to take place via this interface of the Activity Block Container without a need to deal with these messages of the protocol as part of the application design work (*Figures 8a-d*).

With respect to claim **12**, Alford et al. further teaches wherein the Resource Handlers are equipped with an interface for transmitting information between each Resource Handler and the system, this interface being substantially independent of the application session and the resource type (*Figure 7, MessageSend API Function 710*).

With respect to claim **13**, Alford et al. further teaches comprising a dedicated Resource Instance Table in the use of each Resource Handler (*Figure 13a*), and the Resource Handlers are designed to be without intermediate delayed states, wherein the changes of the status data of individual Resource Reservation Instances are stored in the Resource Instance Table of each Resource Handler (*Figure 12*).

With respect to claim **14**, Alford et al. further teaches wherein the Application Session Management and Scheduling means are associated with a Session History Table and the Resource Allocation Manager are associated with a Resource Allocation

Table (*Figures 13b-c*), and the Application Session Management and Scheduling means and the Resource Allocation Manager are designed to be without intermediate states, wherein the changes of session-related status information is stored in said Session History Table, and the changes of session-related information of Resource Reservation Instances are stored in said Resource Allocation Table (*Figure 12*).

With respect to claim **16**, Alford et al. teaches a method for executing application sessions in an electronic device with one or more processors for synchronizing Resource Reservation Instances as well as the execution of substantially simultaneous application sessions (*Figure 10, Scheduling Function 1000*), wherein the application session to be executed comprises one or more Activity Blocks in one or more Activity Block Containers, and an execution order is determined for said Activity Blocks (*Figure 2*), the method comprising at least the following steps:

- a resource management and allocation step for requesting and reserving resources for the application session (*Figure 2, Scheduler 220*),

- a bookkeeping and analysis step for saving and analyzing the resource reservation situation (*Figure 2, Scheduler 220*),

- a scheduling and selection step for selecting the next application session and Activity Block to be executed at least on the basis of said resource reservation situation (*Figure 2, Scheduling Queue 215*),

- an execution step for executing the next Activity Block in the course of the selected application session (*Figure 3, Processor 350*),

wherein in the method, a communication protocol connecting said resource management and allocation step, bookkeeping and analysis step, scheduling and selection step, and the execution step are used to control the execution order and, if necessary, to transfer information between said resource management and allocation step, bookkeeping and analysis step, scheduling and selection step, and execution step (*Figure 2, Internal Msg Source 206 and External Msg Source 205*).

With respect to claim **17**, Alford et al. further teaches wherein a bookkeeping of the resource allocation is maintained, and an first control message is transmitted to an Activity Block to provide control information on the resource allocation at the time of the initiation of the Activity Block, and a second Control message is returned by the Activity Block to provide information about the resources reserved or released by the Activity Block to update the bookkeeping of the resource allocation situation after the completion of each Activity Block (*Figure 3, PTE Application Interface 330*).

With respect to claim **18**, Alford et al. further teaches wherein the resources needed by each Activity Block are reserved and released (*Figure 2, Internal Msg Source 206 and External Msg Source 205*) by the application session, either directly from resource type specific Resource Handlers or from the Resource Allocation Manager that enable the queuing of Reservation Request messages, on the basis of control parameters received in a first control message received from Application Session Management and Scheduling means (*Figure 11*).

With respect to claim **19**, Alford et al. further teaches wherein second control messages are used by the application session to dynamically assign Resource

Reservation Instances to the use of different Activity Block Containers involved in the execution of the session, as needed (*Figure 7*).

With respect to claim **20**, Alford et al. further teaches wherein in the method, an Operating System is utilized comprising task switching functions (*Figure 3*), and that for synchronizing the reservation, release and other resource-related control from the Application Session Management and Scheduling means, Activity Block Containers, Resource Allocation Manager and the Resource Handlers (*Figure 3, PTE 330*), there is a Session Control Protocol composed of application-independent control messages and rules on their use, which is arranged, during its operation, to implement the synchronization and scheduling control of the execution of the Application Session Management and Scheduling means, the Activity Block Containers, the Resource Allocation Manager, as well as the Resource Handlers (RH), on the basis of the task switching functions of the Operating System as well as the OS task priorities defined for the Application Session Management and Scheduling means, the Activity Block Containers, the Resource Allocation Manager, and the Resource Handlers (*Figure 5, API 540*).

With respect to claim **21**, Alford et al. further teaches wherein in the method, a Resource Instance Table is used per each Resource Handler to provide the resource allocation situation to said Resource Allocation Manager (*Figures 13a-b*), and the synchronization of the bookkeeping and analysis step with respect to the resource management and allocation step of the Resource Handlers is determined so that substantially immediately after each execution turn of the Resource Handlers, it is the

turn of the bookkeeping and analysis step, wherein the resource allocation situation is unambiguously known in the Resource Instance Tables regarding the changes occurred (*Figure 12*).

With respect to claim **22**, Alford et al. further teaches wherein the scheduling of the scheduling and selection step with respect to the bookkeeping and analysis step is determined so that the scheduling and selection step is in turn substantially immediately after the execution of the bookkeeping and analysis step, wherein the resource allocation situation is unambiguously known in the Resource Allocation Table regarding the latest changes occurred, and values can be determined by the Application Session Management and Scheduling means for the parameters of the control messages generated by it for the synchronization of the use of various types of Resource Reservation Instances (*Figure 12*).

With respect to claim **23**, Alford et al. further teaches wherein an End State Module is placed at the end of each Activity Block to complete the execution of the block (*Figure 12, End 1270*), and a Waiting State Module is placed in the Activity Block Container holding the Activity Block (*Figure 9, Blocked 930*), and the execution control of the Activity Block Container holding the Activity Block generates a second control message in the End State Module and pauses the execution in the Waiting State Module in order to wait for an first control message from the Application Session' Management and Scheduling means, wherein the execution of the application session is temporarily interrupted regarding the current Activity Block Container (*Figure 11*).

With respect to claim **25**, Alford et al. further teaches wherein the Activity Blocks of the application session are placed in one or more Activity Block Containers, Activity Blocks in any one of these Activity Block Containers are executed temporally at different times, and in the presence of Activity Blocks that are intended to be executed substantially at the same time in the course of the session, they are placed in different Activity Block Containers (*Figure 6*).

With respect to claim **26**, Alford et al. further teaches wherein for designing applications that are to be executed in the system, each Activity Block Container is furnished with an interface module at those points where the execution of an Activity Block or Activity Block Container can be interrupted and it may be the turn of another Operating System task to be executed, thus enabling the sending and reception of Session Control Protocol messages to take place via this interface of the Activity Block Container without a need to deal with these messages of the Session Control Protocol as part of the application design work (*Figures 8a-d*).

With respect to claim **27**, Alford et al. further teaches wherein the Resource Handlers are equipped with an interface for transmitting information between each Resource Handler of the system, this interface being substantially independent of the application session and the resource type (*Figure 7, MessageSend API Function 710*).

With respect to claim **28**, Alford et al. further teaches wherein a dedicated Resource Instance Table is in the use of each Resource Handler (*Figure 13a*), and the Resource Handlers are designed to be without intermediate delayed states, wherein the

changes of the status data of individual Resource Reservation Instances are stored in the Resource Instance Table of each Resource Handler (*Figure 12*).

With respect to claim **29**, Alford et al. further teaches wherein a Session History Table is in the use of the scheduling and selection step, and a Resource Allocation Table is in the use of the bookkeeping and analysis step (*Figures 13b-c*), the resource management and allocation step, bookkeeping and analysis step, as well as scheduling and selection step are designed to be without intermediate delayed states, wherein the changes of session-related status information is stored in said Session History Table, and the changes of session-related information of the Resource Reservation Instances are stored in said Resource Allocation Table (*Figure 12*).

With respect to claim **31**, Alford et al. teaches an electronic device comprising means for executing application sessions, one or more processors (*Figure 3, Processor 350*), and means for scheduling Resource Reservation Instances as well as the execution of substantially simultaneous application sessions (*Figure 2, Scheduler 220*), wherein the application session to be executed comprises one or more Activity Blocks in one or more Activity Block Containers (*Figure 5*), and an execution order is determined for said Activity Blocks (*Figure 2, Scheduling Queue 215*); the electronic device further comprising resource type specific Resource Handlers for reserving resources for the application session (*Figure 2, Threads 231-n*), Resource Allocation Manager for analyzing and saving a resource allocation situation, Application Session Management and Scheduling means for selecting at least the next application session and Activity Block to be executed on the basis of said resource allocation situation (*Figure 2,*

Scheduler 220), executing means for executing the next Activity Block in the course of the selected application session (*Figure 3, Processor 350*); and the electronic device is provided with a protocol connecting the Resource Handlers, Resource Allocation Manager, Application Session Management and Scheduling means and executing means, to control the execution order and to implement the transfer of information between said Resource Handlers, Resource Allocation Manager, Application Session Management and Scheduling means, and executing means (*Figure 3, PTE Application Interface 330*).

With respect to claim **32**, Alford et al. teaches an wireless communication device comprising means for executing application sessions, one or more processors (*Figure 3, Processor 350*), and means for scheduling Resource Reservation Instances as well as the execution of substantially simultaneous application sessions (*Figure 2, Scheduler 220*), wherein the application session to be executed comprises one or more Activity Blocks in one or more Activity Block Containers (*Figure 5*), and an execution order is determined for said Activity Blocks (*Figure 2, Scheduling Queue 215*); the wireless communication device further comprising resource type specific Resource Handlers for reserving resources for the application session (*Figure 2, Threads 231-n*), Resource Allocation Manager for analyzing and saving a resource allocation situation, Application Session Management and Scheduling means for selecting at least the next application session and Activity Block to be executed on the basis of said resource allocation situation (*Figure 2, Scheduler 220*), executing means for executing the next Activity

Block in the course of the selected application session (*Figure 3, Processor 350*); and the electronic device is provided with a protocol connecting the Resource Handlers, Resource Allocation Manager, Application Session Management and Scheduling means and executing means, to control the execution order and to implement the transfer of information between said Resource Handlers, Resource Allocation Manager, Application Session Management and Scheduling means, and executing means (*Figure 3, PTE Application Interface 330*).

With respect to claim **33**, Alford et al. teaches a software program comprising machine executable steps for executing application sessions in an electronic device with one or more processors for synchronizing Resource Reservation Instances as well as the execution of substantially simultaneous application sessions (*Figure 10, Scheduling Function 1000*), wherein the application session to be executed comprises one or more Activity Blocks in one or more Activity Block Containers, and an execution order is determined for said Activity Blocks (*Figure 2*), the software program further comprising machine performing at least the following steps: executable steps for

- a resource management and allocation step for requesting and reserving resources for the application session (*Figure 2, Scheduler 220*),

- a bookkeeping and analysis step for saving and analyzing the resource reservation situation (*Figure 2, Scheduler 220*),

- a scheduling and selection step for selecting the next application session and Activity Block to be executed at least on the basis of said resource reservation situation (*Figure 2, Scheduling Queue 215*),

- an execution step for executing the next Activity Block in the course of the selected application session (*Figure 3, Processor 350*),

wherein the software program also comprises machine executable steps for using a communication protocol connecting said resource management and allocation step, bookkeeping and analysis step, scheduling and selection step, and the execution step to control the execution order and, if necessary, to transfer information between said resource management and allocation step, bookkeeping and analysis step, scheduling and selection step, and execution step (*Figure 2, Internal Msg Source 206 and External Msg Source 205*).

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

Claims **9**, **15**, **24**, **30** are rejected under 35 U.S.C. 103(a) as being unpatentable over Alford et al. (US Patent Publication 2002/0065953 A1). as applied to claims 1, 8, 16, 23 above, and further in view of Isobe Tadaaki (~~Foreign Patent Document~~ JP 09-305268).

Alford et al. teaches claims 1, 8, 16, and 23 as discussed above which are encompassed in claim 9, 15, 24, and 30 but does not specifically teach the matter contained within claims 9, 15, 24, and 30.

Isobe Tadaaki teaches wherein the Application Session Management and Scheduling means are arranged to analyze the resource allocation situation and the scheduling of the sessions to be executed to detect an overload condition of one or more resources and to manage it by replacing, as needed, application sessions with other application sessions requiring less resources, or by delaying, as needed, the transmission of first control messages to the application sessions, which results in a temporary suspension of the ongoing application session or in a delayed initiation of a new application session (*Isobe Tadaaki Abstract*).

It would have been obvious to a person of ordinary skill in the art to combine the invention of Alford et al. with that of Isobe Tadaaki in order to provide better power consumption.

With respect to claim **15**, the combination noted above further teaches comprising means ASM to determine the load condition of the processor and to adjust the power consumption of the processor on the basis of the load condition through the scheduling of the activities of the application sessions (*Isobe Tadaaki Abstract*).

With respect to claim **24**, the combination noted above further teaches wherein the Application Session Management and Scheduling means analyze the resource allocation situation and the scheduling of the sessions to be executed to detect an overload condition of one or more resources and to manage it by replacing, as needed,

application sessions with other application sessions requiring less resources, or by delaying, as needed, the transmission of first control messages to the application sessions, which results in a temporary suspension of the ongoing application session, or in a delayed initiation of a new application session (*Isobe Tadaaki Abstract*).

With respect to claim **30**, the combination noted above further teaches wherein the load condition of the processor is determined, and the power consumption of the processor is adjusted on the basis of the load condition through the scheduling of the activities of the application sessions (*Isobe Tadaaki Abstract*).

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Examiner Jay Marcyes, whose telephone number is 571-270-1733. The examiner can normally be reached on Monday-Thursday and alternate Friday from 7:30 am to 5:00 pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, David Robertson can be reached on 571-272-4186. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Jay Marcyes

04/27/2007

DAVID L. ROBERTSON
SUPERVISORY PATENT EXAMINER